GridPACK Configuration

We expect execution of a GridPACK application to be determined in part by a single input configuration file (perhaps extracted from a database or web service). We assume that a configuration file is a binding of a hierarchically structured set of keys to a set of values. The Configuration module is used to access values and abstract the structure of the configuration file which might be concretely represented in a ways including XML or with a custom parser. Different components within GridPACK will share a common configuration file and we expect then the top-level of the key name space to identify the module that is primarily associated with a configuration parameter. This provides extensibility over time and insulates modules from each other’s internals.

Key are represented by std::string with the character Configuration::KeySeparator,‘.’ (properly escaped in literals) used as a name specifier. So “foo.bar” is a key associated by convention with module “bar”.

Values are typed and may be of the following primitive types: bool, int, double or std::string. We also allow a value to be a std::vector<double>.

We use the term *cursor* to a common prefix on a set of names. Access to a value may be specified by a complete key or by a cursor and the corresponding suffix relative to that cursor. Thus a cursor may represent the prefix “foo” and relative to that cursor the string “bar” selects the same values as “foo.bar”. Aside from factoring out common information for the client, it allows a common configuration substructure do be used by different modules.